Anomalous Trajectory Detection in Vehicular Social Networks with Unsupervised Dynamic Network Representation Learning

Guojiang Shen, Yuwei He, Xiangjie Kong, *Senior Member, IEEE,* Zhanhao Ji, Lei Wang, *Member, IEEE,* Junjie Zhou, and Linglong Hu

***Abstract*—As data volumes surge, the automatic detection of anomalous trajectories has become increasingly difficult in real-world scenarios. However, most existing works fail to account for the spatial-temporal properties of trajectory simultaneously and rely on biased real-world or synthetic datasets. To tackle these challenges, we propose a novel unsupervised method, Dynamic Network Representation learning (DNR), that leverages the advances in the network field and adapts them to transportation. DNR constructs a dynamic network from the perspective of vehicle social networks using the idea of the k-nearest neighbors. A dynamic walk algorithm is proposed to sample nodes and context over time, focusing on topology changes. DNR embeds the network using Skip-Gram network representation and incremental learning to obtain two trajectory embeddings for two anomaly types. Our experiments on four real-world datasets and eight generative datasets demonstrate the effectiveness of DNR over other methods and reveal its rationality to detect deviations from typical trajectories.**

***Index Terms*—Anomalous trajectory detection, unsupervised learning, vehicular social network, dynamic network representation, typical trajectory.**

I. Introduction

THE transportation industry has been revolutionized by the proliferation of mobile devices and sensor technologies, which has resulted in a significant increase in the collection of trajectory data from vehiculars. This data can be utilized to gain valuable insights into traffic patterns, route planning, and other transportation-related issues. However, with the surge in data volume, the identification of anomalous trajectories that signify the taxi drivers who make unnecessary detours and drive at inappropriate speeds, or result in other hazardous conditions has become increasingly difficult [1]. Thus, the automatic detection of anomalous trajectories has emerged as a critical issue in real-world scenarios.

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2023C01241, in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003, and in part by the National Natural Science Foundation of China under Grant 62072409 and Grant 62073295. *(Corresponding author: Xiangjie Kong.)*

Guojiang Shen, Yuwei He, Xiangjie Kong and Zhanhao Ji are with College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: gjshen1975@zjut.edu.cn;211122120044@zjut.edu.cn;xjkong@ieee.org; 2112012108@zjut.edu.cn).

Lei Wang is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: l-wang@buaa.edu.cn).

Junjie Zhou and Linglong Hu are with Zhejiang Supcon Information Co., Ltd., Hangzhou 310052, China. (e-mail: zhoujunjie@supconit.com; hulinglong@supconit.com).

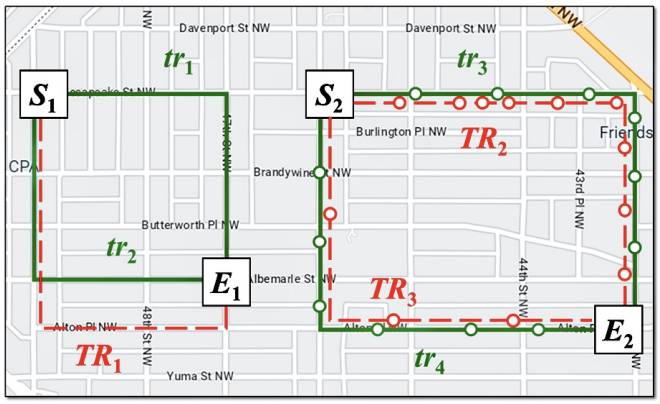


Fig. 1. Examples of anomalous trajectory detection.S_1 and S_2 are the start points of the trajectories, E_1 and E_2 are the end points of the trajectories, TR_1 on the left shows the spatial trajectory anomalies, and TR_2 and TR_3 on the right show the time trajectory anomalies.

In recent years, anomalous trajectory detection has attracted widespread attention. Researchers have proposed various approaches for detecting anomalous trajectories, including statistical analysis [2]–[5], machine learning [6]–[9], and deep learning methods [10]–[17]. The fundamental concept underlying existing methods is to identify trajectories that significantly deviate from normal traffic patterns and behaviors. Generally, a trajectory is considered anomalous if it deviates from the typical travel path of a route. For example, Wang *et al.* [4] identify the typical trajectory and calculate anomaly scores by set operation. However, they only consider the spatial aspect of trajectories and overlook their temporal properties. Additionally, some previous work [2], [6]–[8] rely on handcrafted features, which can introduce bias in anomaly detection. Moreover, Liu *et al.* [12], [18] attempt to detect potential anomalies in an unsupervised manner. However, their approach requires training on a clean dataset, which needs manually prelabeled, or otherwise be impacted by a dirty dataset.

In general, existing methods for detecting anomalous trajectories encounter two significant challenges:

- **Imbalance in properties.** Trajectory anomaly detection involves considering both spatial and temporal aspects, given that trajectories are inherently spatial-temporal data, as shown in Figure 1. However, many existing methods tend to overly emphasize typical routes and sparse trajectories that deviate from spatial clusters, while overlooking the temporal

properties. Consequently, there is a need to simultaneously account for the spatial-temporal properties, that is, focusing on the variation of the spatial properties over time.

- **Imbalance in datasets.** Some methods rely solely on either manually labeled or anomaly-injected datasets to evaluate their performance. Manually labeled datasets are labor-intensive to construct and their credibility may be questionable, while anomaly-injected datasets cannot guarantee the absence of anomalous trajectories prior to injection. Thus, relying on a single type of dataset can introduce bias into the evaluation process.

To simultaneously observe the spatial and temporal properties of trajectories, a novel approach is needed. Fortunately, dynamic network representation learning [19] has rapidly advanced, enabling the incorporation of temporal features into static network representations [20]. However, its application to transportation requires further adaptation, such as *constructing dynamic vehicular networks* and *sampling dynamic networks*. Inspired by prior works, this paper proposes a method for anomalous trajectory detection based on dynamic network representation learning, named DNR. Skip-gram network representation learns word embeddings by maximizing the probability of predicting the word in context. Therefore, DNR is an unsupervised learning approach that learns the representation of nodes without relying on the label information of the nodes. The method is starting with the social properties of vehicles at each moment and focuses on the cluster relationships of vehicles over time, constructs dynamic vehicular social networks using the idea of k-nearest neighbors [21]. Then, the dynamic walk algorithm is designed to sample dynamic networks. Finally, the method employs Skip-Gram network representation [22] to network node embeddings, resulting in two trajectory embeddings for evaluation: driving embedding and average transfer embedding, which can identify two types of anomaly.

To address the dataset limitations associated with existing methods, we employ a combination of manually labeled and anomaly-injected datasets. Initially, volunteers are engaged to annotate trajectories with identical start and end points, forming the original experimental dataset. Subsequently, trajectories labeled as normal are selected, and a specialized code is utilized to generate abnormal trajectories, accounting for both temporal and spatial anomalies. By merging the normal and anomalous trajectories, the temporal anomaly datasets and detour anomaly datasets are created. Then, extensive experiments are conducted to evaluate the proposed method, which not only validates its effectiveness and advantages compared to alternative methods but also demonstrates the ability of dynamic network representation learning in identifying typical trajectories. These findings provide a strong rationale for the application of our method in anomaly trajectory detection.

Overall, our main contributions of this paper can be summarized as follows:

- We propose a novel unsupervised method, named DNR, leveraging the power of dynamic network representation learning for detecting anomalous trajectories in the absence of prior knowledge. It constructs dynamic vehicular social networks based on a simple yet effective approach.
- We design a dynamic walk algorithm and incorporate incremental learning to optimize the sampling process for the dynamic nature of the vehicular social network.
- Comprehensive experiments are conducted on both manually labeled and anomaly-injected datasets to validate the effectiveness of DNR on spatiotemporal features. Furthermore, we demonstrate the rationality of DNR to detect deviations from typical trajectories.

The rest of the paper is organized as follows. Section 2 review related work. Our method is presented in Section 3. Section 4 shows the experimental setup, results, and discussion of the findings. Finally, Section 5 concludes the paper.

II. Related Work

A. Anomalous Trajectory Detection

Among the statistical-based methods, trajectory clustering is a key method for detecting anomalous behavior. The rich geospatial trajectory data provides a strong foundation for spatial anomaly detection. Zhang *et al.* [2] proposed the incremental behavior anomaly detection (iBAT) method, which models taxi behavior by using temporal and spatial information. It is a classic method that laid the foundation for many subsequent works. Lv *et al.* [3] proposed DBOTD that analyzes multiple features of trajectories and uses DBSCAN for trajectory clustering to capture outlier trajectory features. Wang *et al.* [4] proposed an anomalous trajectory detection and classification (ATDC) method based on difference and intersection distance. However, these clustering-based statistical methods rely on the selection of clustering algorithms and parameters setting. Subtle variations in these selections can yield vastly different detection results, which can affect the reliability and robustness of the method.

In addition, some studies focus on global trajectories/sub-trajectories, aiming to detect various types of anomalies by analyzing urban traffic information and combining temporal and spatial dimensions. Kong *et al.* [10] proposed the long-term traffic anomaly detection (LoTAD). It partitions and clusters trajectories into different road sections, and analyzes their point distribution to build distribution models. Zhu *et al.* [7] used time-dependent popular route information (TPRO), which cluster trajectories and calculate anomaly scores of clusters. To tackle online detection, they further proposed the time-dependent popular route-based online trajectory anomaly detection [23] method, which groups trajectories based on pedestrian travel habits to extract time-dependent popular routes, and detects anomalous with grouped trajectory anomaly detection algorithm. For sub-trajectory processing, Chen *et al.* [6] proposed an online anomalous trajectory detection method called iBOAT. It converts the anomaly detection problem into a binary classification problem. Mao *et al.* [8] proposed a trajectory data stream-based method that segments data into batches, obtains representations through feature aggregation, and detects outliers using a density-based method. Yang *et al.* [9] detect anomalous points in trajectories using MiPo, a tabular anomaly detector. Trajectory data is represented as a table with multiple attributes and processed with the table

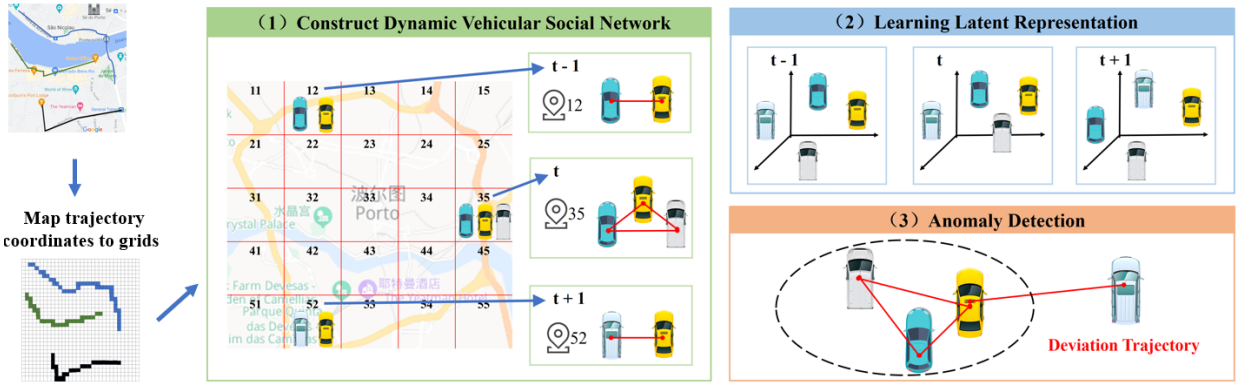


Fig. 2. Our model consists of three main steps. First, the coordinates of vehicle trajectories are mapped to the corresponding grid, meaning that at each time step, the vehicle trajectory coordinates are projected onto the respective grid locations. Next, vehicles are abstracted as nodes at each time step, and a social network of vehicles is constructed accordingly. By integrating networks from different time periods, a dynamic network structure is formed. Subsequently, a dynamic walking method is employed to learn the representation of the network. Finally, anomaly scores are calculated based on the learned representations to detect abnormal behaviors.

anomaly detector. However, these methods may incorrectly identify normal points as abnormal points, and overlook temporal evolution features in the upstream and downstream of local trajectories, which can result in missed detections and false positives.

B. Unsupervised Learning

Deep learning utilizes neural networks to identify the mapping function that fits the regularities of a large amount of data, making it ideal for detecting anomalous trajectories. Cheng *et al.* [11] proposed an STRNN for modeling and learning trajectory data as spatiotemporal sequence data to capture the spatial and temporal correlations in trajectories. However, this method relies on supervised learning, requiring considerable human effort in identifying normal or anomalous trajectories before training. On the other hand, Liu *et al.* [12] proposed the OGSM method for unsupervised learning based on deep generative sequence models. It uses recurrent neural networks and variational autoencoders as encoder and generator, respectively, to learn the encoding of historical trajectories. Wang *et al.* [14] proposed an attention-based deep embedding method which uses the VAE model to learn trajectories. However, these method [12] [14] assume all trajectories in the dataset to be normal during training, and [12] only detects incremental trajectories as anomalous during testing, which limits its applicability to unlabeled trajectory datasets. In addition, there are some vision-based anomaly detection methods [13] that can provide valuable insights, but they suffer from similar limitations. Therefore, although neural networks are powerful, they demand high-quality labeled datasets, making them less applicable to unlabeled trajectory datasets.

C. Network Representation Learning

Representation learning [24] also has applications in trajectory anomaly detection. Network representation learning [20] is a powerful method for feature extraction and learning from network data. It aims to represent nodes and edges of a network as low-dimensional vectors, enabling efficient analysis and prediction. Unlike traditional machine learning methods, network representation learning can handle complex topologies, such as social networks [25], biological networks [26] and transportation networks [27]. One of the key challenges in network representation learning is to efficiently sample a subset of the network while maintaining its structural properties.

One approach to achieve this is through random walk-based sampling methods. These methods start at a randomly chosen node and traverse the network by following a sequence of edges based on specific rules. However, different methods have different concerns, which can affect the representation of learning results. Random walk [28] is a common method, where the network is traversed by randomly selecting an edge at each node. This method can efficiently explore the network and generate a large number of walks, but it may not capture the network's structural properties. To address this limitation, several modified walk methods have been proposed, such as DeepWalk [29], Node2Vec [30] and LINE [31]. These biased random walk methods use a transition probability matrix to bias the selection of edges based on desired criteria, such as node degree or community membership. This allows for better exploration of the network's local structure and can improve the quality of the generated walks.

In recent years, network representation learning has been extended to handle temporal aspects, leading to the development of dynamic network representation learning [19]. This method enables the modeling of networks with spatiotemporal properties and the development of models capable of handling such features. By incorporating time into the network representation, researchers can examine how the network evolves over time and mine information accordingly.

III. Methodology

In this paper, we propose a novel unsupervised method to detect anomalous trajectories. The framework of our proposed method can be seen in Figure 2. It consists of three main steps. Firstly, the vehicular trajectory coordinates are mapped to the corresponding grids, thus at each time step, vehiculars are

abstracted as nodes and constructed as the vehicular social network. Multiple networks at various times constitute a dynamic network. Then, the network is used to learn the representations. Among them, a dynamic walk method is proposed, which will be presented separately due to its complexity. Finally, the obtained representation is used to compute anomaly scores to detect anomalies. The detail of our method will be discussed in the following parts.

Algorithm 1 Vehicular Trajectory Processing

Require: Coordinates set of trajectory, Tr_c;
Ensure: Grids set of trajectory, Tr_g;
1: Initialize $Tr_g = \emptyset$
2: Retrieve the latitude and longitude ranges of Tr_c, where $maxLng$, $minLng$, $maxLat$ and $minLat$ denote the maxima and minima of longitude and latitude, and $lngS$ denotes how much longitude each grid represents, $latS$ denotes how much latitude each grid represents. The trajectories are processed by dividing the longitude and latitude directions into 100 grids each.
3: $lngS = (maxLng - minLng) \ / \ 100$
4: $latS = (maxLat - minLat) \ / \ 100$
5: **for** tr_i in Tr_c **do**
6: **for** $c_i^t = (lat_i^t, \ lng_i^t, \ \dots)$ in tr_i **do**
7: $x =$ **ceil**$((lng - minLng) \ / \ lngS)$
8: $y =$ **ceil**$((lat - minLat) \ / \ latS)$
9: $Tr_g[i][t] = 10^5 x + y$
10: **end for**
11: **end for**

A. Dynamic Vehicular Social Network

*1) **Vehicular Trajectory Processing**:* Before building a dynamic social network of vehicular trajectories, the first step is to map trajectory coordinates to grids based on the k-nearest neighbors' idea, as outlined in Algorithm 1. This involves dividing the map into specific grid sizes and mapping the trajectory coordinates accordingly. We traverse each trajectory tr_i in coordinates set of trajectory Tr_c, and the trajectory points c_i^t in tr_i which is a time sequences, respectively, as shown in step 5 and 6 of Algorithm 1, and then obtain the grid number $Tr_g[i][t]$ of the trajectory point c_i^t by step 7 and 8, and finally obtain grids set of trajectory Tr_g.

*2) **Dynamic Social Network Construction**:* The process of constructing the dynamic social network is shown in Algorithm 2. To construct a dynamic social network of vehicular trajectories, the first step is to construct a static social network for each time stamp, which deals with spatial data, as shown in step 4 to 10 of Algorithm 2. The vehicular trajectory comprises a series of coordinates, where each coordinate corresponds to a particular moment in time. By treating each coordinate as a network node (represented by v), the set of all coordinates becomes the set of network nodes (denoted by $\mathcal{V}$) at the moment, containing a total of $|\mathcal{V}|$ nodes. Specifically, given a specific time t, we traverse the g_i^t in each trajectory grids sequence g_i in the grids set of trajectory (Tr_g), set the set of vehicle nodes $d[g_i^t]$ for each grid number g_i^t, and add the vehicle nodes v_i to the set of vehicle nodes $d[g_i^t]$ that corresponds to its grid number g_i^t at this moment. It is implemented in step 5 and 9 of Algorithm 2. In order to establish connections between nodes, it is essential to determine whether there is an edge between any two pairs of nodes (v_i, v_j) in $\mathcal{V}$.

In brief, the resulting trajectories are represented as a chronological sequence of grid numbers by Algorithm 1. If two coordinates, v_i and v_j, fall into the same grid, i.e., v_i and v_j have the same grid number, a connection edge between them is established, i.e., $e_{ij} = 1$. Otherwise, $e_{ij} = 0$. By identifying and connecting each node pair, which is implemented in step 6 to 7 of Algorithm 2, the current vehicular social network can be obtained. The set of connecting edges is denoted as $\mathcal{E}$ and comprises a total of $|\mathcal{E}|$ edges. As a result, spatial properties data are processed.

By treating the static social network at different time steps as a set, the corresponding dynamic social network can be derived, which reflects temporal dynamics, i.e., utilizes the temporal property. Since a trajectory is a finite set, the dynamic social network of vehicular trajectories is denoted as $\mathcal{G}^T = (G^1, G^2, \dots, G^T)$, as shown in step 11 of Algorithm 2, where T represents the maximum time value among all trajectories. As a result, the time property is utilized.

The continuous change in trajectory coordinates over time leads to dynamic edge additions or removals and topological changes within the vehicular social network. Initially, the connected edges tend to be dense at the beginning of vehicular departure, but as vehiculars move forward and select varying speeds and routes, the network becomes relatively sparse. This results in the division of the vehicular social network into several disconnected sub-networks, denoted as $G^t = G_1^t \cup G_2^t \cup \cdots \cup G_S^t$, where S represents the number of sub-networks.

Algorithm 2 Dynamic Social Network Construction

Require: Grids set of trajectory, Tr_g;
Ensure: Dynamic social network, $\mathcal{G}^T$;
1: $T =$ **max_time**(Tr_g)
2: Fill each trajectory in Tr_g with its last grid number to achieve a length of T
3: **for** t in $\{1, \ 2, \ \dots, \ T\}$ **do**
4: Initialize G^t as empty network and $d = \emptyset$
5: **for** $g_i = (g_i^1, \ g_i^2, \ \dots, \ g_i^T)$ in Tr_g **do**
6: **for** j in $d[g_i^t]$ **do**
7: G^t.add_edge$(i, \ j)$
8: **end for**
9: $d[g_i^t]$.add(i)
10: **end for**
11: $\mathcal{G}^T[t] = G^t$
12: **end for**

B. Dynamic Walk

In a dynamic social network, vehiculars that travel along typical trajectories are likely to remain in the same subnetwork for extended periods. Therefore, in contrast to vehicular social networks that prioritize the overall topology, dynamic social networks should prioritize changes in the connected edges of vehiculars and update the topology accordingly. However, sampling node-to-node contextual information

can be challenging. Selectively sampling nodes from G^t may lead to biased results that favor nodes with high-density sub-networks. One alternative is to sample all nodes, which may better preserve the network topology but can be too burdensome for the constructed network. To address this issue, this paper propose the dynamic walk (DW) method inspired by dynamic network embedding [19] and GloDyNE [32]. The method focuses on capturing topology changes while avoiding the need for full network sampling at all times. It comprises three components: network partitioning, dynamic network node selection, and a sampling method. The latter two components reflect temporal proximity and spatial proximity, respectively.

*1) **Network Partitioning**:* Firstly, network partitioning algorithm is used to regulate the density of the sub-networks, thus improving the computational quality of the node selection strategy. Currently, the most common objective function in network partitioning algorithms [33] is minimizing edge cut.

$$\min \sum_{1\le m<n\le P} \{e_{i,j} | v_i \in \mathcal{V}_m, v_j \in \mathcal{V}_n, e_{i,j} \in \mathcal{E}\}, \quad (1)$$

where i and j denote the node IDs, while m and n denote the sub-network IDs. $\mathcal{V}_m$ and $\mathcal{V}_n$ are the node sets of the corresponding sub-network, and P represents the number of sub-networks obtained after partitioning the network. The binary variable $e_{i,j}$ indicates the presence of an edge, with a value of 1 if the edge exists and 0 otherwise. At this point, $\mathcal{V}_m \cap \mathcal{V}_n = \emptyset$, and $\mathcal{V} = \bigcup_p \mathcal{V}_p$. Minimizing the objective function increases the likelihood of partitioning high-density sub-networks.

In practice, the classic METIS algorithm [34] is selected for network partitioning. It is a hierarchical partitioning algorithm that consists of three main steps. Firstly, it sparsified the network using an edge-weighted priority matching strategy, recursively reducing the size of the original network by collapsing nodes and edges. This transforms the network into a series of smaller abstract networks. Secondly, when the network is reduced to a certain size, the K-way partitioning algorithm is applied. It initializes nodes randomly and performs a breadth-first search to obtain the sub-network with the minimum cut edge. Finally, by recursively expanding the partitioned sub-networks and minimizing Equation (1), G^t is partitioned into P sub-networks, denoted as $G_1^t, G_2^t, \ldots, G_P^t$.

*2) **Dynamic Network Node Selection**:* Secondly, in order to address the dynamic nature of the network, it is essential to select a representative node from each partitioned sub-network to ensure diversity on G^t. However, selecting representative nodes from these sub-networks is a challenging task. Fortunately, prior research [35] has studied the evolution of network structure and proposed measures for assessing the speed and scale of network structure evolution. Building upon this work, we propose a score function that evaluates the importance of nodes based on their inbound and outbound links, as shown below.

$$\begin{cases} \mathrm{S}(v_i^t) = |\Delta\mathcal{E}_i^t| + \mathcal{R}_i^{t-1} \\ |\Delta\mathcal{E}_i^t| = \left|\mathcal{N}(v_i^t) \cup \mathcal{N}(v_i^{t-1}) - \mathcal{N}(v_i^t) \cap \mathcal{N}(v_i^{t-1})\right| \end{cases}, \quad (2)$$

where $|\Delta\mathcal{E}_i^t|$ is determined by computing the set operations of intersection, union, and difference on the neighbor nodes of v_i at two adjacent times, which represents the number of edges connected to v_i that have changed at the current time t. $\mathcal{R}_i^{t-1}$ is the cumulative change count of node v_i up to time $t-1$. When v_i is selected as a representative, its cumulative change count $\mathcal{R}_i^{t-1}$ is set to zero, while it continues to grow otherwise.

When a node v_i^{t-1} undergoes a change in its connected edges and enters another sub-network, its score $\mathrm{S}(v_i^t)$ will be higher than that of an invariant node within the same sub-network. This is due to its higher number of changed edges $|\Delta\mathcal{E}_i^t|$. Consequently, v_i^t is more likely to be chosen as a representative node.

Then, by applying the Softmax function on the topological change score within the cut sub-network G_p^t, the score is normalized to obtain the selection probability:

$$\mathrm{P}(v_i^t) = \frac{e^{\mathrm{S}(v_i^t)}}{\sum_{v_j^t \in \mathcal{V}_p^t} e^{\mathrm{S}(v_i^t)}} \quad v_i^t \in \mathcal{V}_p^t, \quad (3)$$

where $\mathcal{V}_p^t$ represents the set of nodes in G_p^t. The equation guarantees that a node will have a non-zero normalization probability even if its topological change score is 0. Moreover, if there are no changes within the sub-network, a representative node will be selected with equal probability from all nodes in $\mathcal{V}_p^t$, as shown in step 11 of Algorithm 3.

Algorithm 3 Dynamic Network Node Selection

Require: Selection parameter, α; Dynamic social network, $\mathcal{G}^T = \{G^1, G^2, \ldots, G^T\}$;
Ensure: Node selection set at time t, $\mathcal{V}_{sel}^t$;

```
1: Initialize R^0 = ∅
2: for t in {1, 2, ..., T} do
3:   if t = 1 then
4:     V_sel^1 = V^1
5:   else
6:     Calculate |ΔE^t| within G^{t-1} and G^t through Equation (2)
7:     R_i^t = |ΔE_i^t| + R_i^{t-1}
8:     P = α|V^t|
9:     Partition G^t using METIS algorithm into P sub-networks
10:    for p in {1, 2, ..., P} do
11:      Randomly select a representative node using Equation (3) and add it to V_sel^t
12:    end for
13:    Set the value of R_i^t to 0 for all nodes in V_sel^t
14:  end if
15: end for
```

In brief, each timestamp dynamic social network G^t is divided into P sub-networks according to the METIS algorithm, and then representative nodes are selected in each sub-network according to Equation (2) and Equation (3). Specifically, P is set to $\alpha|\mathcal{V}^t|$, where α is a parameter that controls the number of nodes to be selected. Note that Equation (2) cannot be applied to nodes at the initial time, and therefore, all nodes will be

seleted as representative nodes. The dynamic network node selection algorithm is outlined in Algorithm 3.

The algorithm obtains the set of all nodes in G^1 at $t = 1$, and selects the set of nodes in G^t with probability at $t \geq 1$, which is collectively called $\mathcal{V}_{sel}^t$.

*3) **Sampling Method**:* To obtain contextual information for nodes in $\mathcal{V}_{sel}^t$, a sampling method is necessary to explore the network topology. Various random walking algorithms, such as RandomWalk [28] and DeepWalk [29], have been proposed to sample the network. In this work, the random walk algorithm is employed. Specifically, for each node in $\mathcal{V}_{sel}^t$, the algorithm assigns equal probabilities to its neighboring nodes and samples a topology of length l. The resulting contextual information set, denoted as $\mathbf{V} \in \mathbb{R}^{|\mathcal{V}_{sel}^t| \times n \times l}$, includes $|\mathcal{V}_{sel}^t|$ results, each of which has n contextual information instances consisting of l nodes. It is important to note that the dynamic network node selection algorithm only affects the sampling method through $\mathcal{V}_{sel}^t$, which simplifies the incorporation of the selection algorithm into the random walking process.

From a certain point of view, our method considers the global network features, but the randomness of the random walk still has some impact on the stability of anomaly detection. The random walk algorithm may lead to uneven sampling of nodes and may sometimes miss some important nodes or paths, thus affecting the stability and accuracy of the algorithm. In subsequent studies, we will use the degree of nodes or other importance measures to guide node selection or limit the step size or direction of the random walk, making it more likely that important nodes will be selected. However, these may further increase the computational complexity.

C. Representation Learning

After obtaining the contextual information of the sampled nodes, a sliding window of size $s + 1 + s$ is used to generate positive sample pairs $(v_{center}^t, v_{center+i}^t)$, where v_{center}^t represents the node at the center of the sliding window, $i \in [-s, +s]$ and $i \neq 0$. This sliding window technique is commonly used in word embedding methods such as Word2Vec [36] and has been adapted to network embedding techniques. The positive sample pairs generated from the sliding window constitute the set $\mathcal{D}^t$.

Subsequently, the Skip-Gram Negative Sampling (SGNS) [22] model is employed to learn embedding vectors of the sampled nodes. The objective of SGNS is to maximize the log-likelihood of predicting the context nodes given the center node in the sampled pairs of $\mathcal{D}^t$. This is achieved by minimizing the negative log-likelihood of the observed context nodes and randomly sampled negative nodes. The learning objective is as follows:

$$\max \log \sigma \left(Z_{v_i^t} \cdot Z_{v_p^t} \right) + q \cdot \mathbb{E}_{v_n^t \sim P_{\mathcal{D}^t}} \left[\log \sigma \left(-Z_{v_i^t} \cdot Z_{v_n^t} \right) \right], \tag{4}$$

where $Z_{v_i^t}$ represents the vector representation of vehicular v_i at time t in $\mathbb{R}^d$, where d denotes the dimensionality of the vector. $Z_{v_p^t}$ is the vector representation of positive sample node v_p^t, which is selected from the positive sample pair (v_i^t, v_p^t) in $\mathcal{D}^t$. The negative sample node v_n^t is chosen from the distribution $P_{\mathcal{D}^t}$ based on v_i^t, with q being the number of negative samples. $\sigma(\cdot)$ represents the Sigmoid function.

Besides, an online incremental learning approach is adopted to encode the node embeddings, fully utilizing the characteristics of topology changes in dynamic networks. The node embeddings are increamentally updated as edges are added to or removed from the network. The learning process can be defined as

$$\mathbf{Z}^t = \begin{cases} f\left(G^1, \mathrm{SGNE}\left(\mathbf{Z}_{rand}^1, \mathrm{DW}\left(G^1\right)\right)\right), & t = 1 \\ f\left(G^t, \mathrm{SGNE}\left(\mathbf{Z}^{t-1}, \mathrm{DW}\left(G^{t-1}, G^t\right)\right)\right), & t > 1 \end{cases} \tag{5}$$

where the initial embedding vector $\mathbf{Z}_{rand}^1$ for all vehiculars at $t = 1$ is randomly initialized and updated based on the sampled contextual information of all nodes. For $t > 1$, the model incrementally updates embedding vector $\mathbf{Z}^t$ by probabilistically sampling nodes with high change scores, based on $\mathbf{Z}^{t-1}$. This approach avoids learning from the initialization stage and inefficient global learning. The final node embedding vetor $\mathcal{Z} = \{\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^T\}$ for all times is obtained.

D. Anomaly Detection

This subsection introduces two trajectory embeddings, the driving embedding and the average transfer embedding, based on the node embedding vector of the dynamic network.

To account for potential differences in the magnitude of the node embedding vectors at different times, a normalization step is applied to the vectors

$$\mathbf{Z'}^t = \frac{\mathbf{Z}^t}{\|\mathbf{Z}^t\|_1}, \tag{6}$$

where $\|\cdot\|_1$ denotes the L1 norm. The normalized vectors constitute $\mathcal{Z}'$. Although the dynamic social network is constructed by filling all trajectories to the maximum time as Algorithm 2, each trajectory has its own end time t_i.

vehiculars may remain or leave sub-network where they are located, depending on the drivers. Therefore, the driving embedding is defined as the sum of the normalized embedding vectors of the trajectory during travel, i.e.,

$$e_{d,i} = \sum_{t=1}^{t_i} \mathbf{Z'}_i^t. \tag{7}$$

On the other hand, the average transfer embedding is defined as the average of the difference between the normalized embedding vectors as the trajectory transition between different sub-networks during traveling, i.e.,

$$e_{t,i} = \frac{\sum_{t=2}^{t_i} \mathbf{Z'}_i^t - \mathbf{Z'}_i^{t-1}}{t_i}. \tag{8}$$

By computing the above two equations for each trajectory, the driving embedding $\mathbf{E}_d$ and average transfer embedding $\mathbf{E}_t$ are obtained. To detect anomalous trajectories, the local outlier factor (LOF) [37] algorithm is used. LOF is a density-based outlier detection method that calculates the degree of anomaly for each data point based on the density of its surrounding neighborhood. Using LOF, the driving anomaly score $\mathbf{S}_d$ and the average transfer anomaly score $\mathbf{S}_t$ are

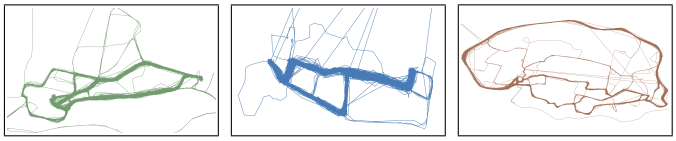
evaluated respectively. Depending on which score is used for anomaly evaluation, this method is referred to as DNR-D or DNR-T.

IV. Experiments

A. Datasets

This section uses a taxi trajectory dataset provided by Kaggle[1], which contains over 1.71 million trajectories records from 442 taxis operating in Porto, Portugal, between January 7, 2013, and June 30, 2014. Each trajectory includes GPS coordinates reported every 15 seconds by the taxi. The trajectories are categorized based on their starting and ending points, and three starting and ending points with high traffic volume are chosen as the dataset, namely the Campanhã Train Station to São Ildefonso Church, São Bento Station to Santa Maria Hospital, and Porto Shopping Center to Campanhã Train Station. To facilitate the subsequent description, these three datasets are named D1, D2, and D3, and their visual effects are shown in Figure 3.

To further validate the model, we also validate it on the Chengdu dataset, which is another large cab trajectory dataset, from August 3, 2014 to August 30, 2014. We choose a starting and ending points with high traffic volume as the dataset and named it D4, which is not visualized due to space limitation.



(a) D1 (b) D2 (c) D3

Fig. 3. Visualization of trajectories for three datasets.

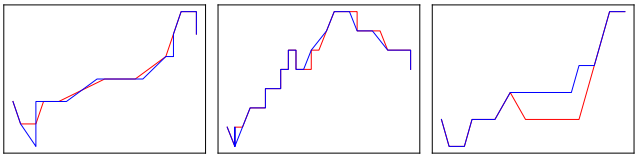
The trajectory characteristics of the four datasets are calculated using a unified approach. The calculation process takes into account the reporting interval provided by the dataset and uses a precise method to calculate the results, which is shown in Table I.

TABLE I
DATASET TRAJECTORY CHARACTERISTICS

Dataset	Number	Minimum Time (Seconds)	Maximum Time (Seconds)	Minimum Distance (Meters)	Maximum Distance (Meters)
D1	761	285	2175	2016	14954
D2	877	285	2835	1872	22962
D3	738	345	3210	4599	19752
D4	1026	200	3405	1069	25764

As done in previous work [4], we ask volunteers who are very familiar with the road network to manually label the four datasets as normal or abnormal to create the basic dataset for the experiments. The evaluation criteria include factors such as trajectory length and shape. However, manually labeled datasets may have credibility issues. To address this, the subset of trajectories labeled as normal is first selected from the dataset. Then, eight additional abnormal datasets are constructed by injecting two types of anomalies separately using a code-based approach. Figure 4 illustrates three examples.

[1]https://kaggle.com/competitions/pkdd-15-predict-taxi-service-trajectory-i



(a) Short-duration (b) Long-duration (c) Detour

Fig. 4. Visualization of injected anomalous trajectories of different types.

Trajectory data is a type of spatio-temporal traffic data with both temporal and spatial properties. Therefore, two types of anomalies, referred to as temporal anomalies and detour anomalies, are specially injected into the subset, i.e., the normal trajectories.

Temporal anomalies. Considering the temporal properties of trajectory data, we modify durations to introduce anomalies. The shortest 4% of normal trajectories are scaled down to 70-80% of their original duration using quadratic interpolation, as shown in Figure 4a. The longest 1% are extended to 150-200% of their original duration, as shown in Figure 4b. The chosen ratio is based on the fact that extending trajectories disrupt spatial properties more significantly than shortening them. Consequently, a dataset is constructed wherein up to 5% of trajectories exhibit temporal anomalies.

Spatial anomalies. Trajectory has spatial properties, which can be disrupted by offsetting trajectory segments. Specifically, 5% of randomly selected trajectories are chosen from the subset, and 30%-50% of trajectory segments are randomly selected for each trajectory. The selected segments are then shifted 100 meters to the left or right of the direction of travel, as shown in Figure 4c. The offset segments exclude the start and end points of the trajectories. Finally, the dataset with 5% spatial anomalies injected is obtained based on the subset.

In the following sections, D1 will be used to refer to the first manually labeled dataset, TD1 will refer to the temporal anomaly dataset generated based on D1, and DD1 will refer to the detour anomaly dataset generated based on D1.

B. Experiments Settings

Metrics. This paper employs the same three metrics as in the previous work [5]: precision, recall, and F1 score. Additionally, the area under the precision-recall curve (PR-AUC) [38] is used as a metric because the proposed method produces the anomaly score for each trajectory. The score range for all metrics is from 0 to 1, with higher scores indicating better results. It is important to note that the focus of this study is on identifying anomalous trajectories, not classification. As anomalous trajectories account for a small proportion of the dataset, some metrics may still be considered good even if they do not reach 0.5.

Baseline. To verify the framework's effectiveness, our paper investigates several unsupervised methods which are as follows:

1) iBAT [2]: This classic method detects anomalies by comparing the extent to which the target trajectory can be

separated from reference trajectories with the same start and end points.

2) DBOTD [3]: This density-based method clusters trajectories with the same start and end points using DBSCAN and calculates the anomaly score of the target trajectory by computing the distance between the target trajectory and the closest typical trajectory.
3) TPRO [23]: This method selects the most typical k trajectories as reference trajectories based on time and calculates the anomaly score of the target trajectory based on them.
4) ATDC [4]: This method evaluates the similarity between any two trajectories using the set difference and intersection, designs an anomaly scoring function to differentiate different types of anomalous trajectories, and discovers different anomaly patterns based on this.
5) MiPo [9]: This method proposes a feature extraction method that extracts tabular features from trajectory data, represents variable-length trajectories as fixed-length tabular data, and endows tabular anomaly detectors with the ability to detect trajectory anomalies.

Parameters. In the Porto dataset, the parameter α for network partitioning is set to 0.1, while n and l are set to 10 and 15, respectively, for the sampling method. As for network representation learning, the hidden state size d of the neural network is set to 32. In the Chengdu dataset, the parameter α for network partitioning is set to 0.12, while n and l are set to 12 and 15, respectively, for the sampling method. Time interval between trajectory points is 15 seconds in the Porto dataset and 12 seconds in the Chengdu dataset. The reasons behind these parameter settings will be analyzed in detail in subsequent sections, we just specifically analyze the Porto dataset in detail due to paper length constraints.

C. Dataset Analysis

To understand the Porto datasets, we calculated and analyzed the travel time and distance of each trajectory, removing the top 5% of anomalous values to avoid skewing the visualization. Figure 5 visualizes the remaining 95% of data, with the Y-axis showing the proportion of trajectory features within each interval.

The figure shows that travel time and distance on D1 dataset and D2 dataset are concentrated within a certain range, resembling a Gaussian distribution. For D3, the travel time is similar to the first two datasets, but the distance is divided into two categories, indicating the presence of two completely different driving routes in the dataset that are far apart in distance but very close in time. These two trajectory routes can be observed in Figure 3c. Additionally, as shown in Figure 3, some trajectories in three datasets far exceed the normal range, indicating the existence of anomalous trajectories.

D. Anomaly Detection

Original dataset. For the four original datasets, the proposed method calculates anomaly scores using driving embeddings, referred to as DNR-D. Trajectories with anomaly scores ranking in the top 10% are considered anomalous trajectories for

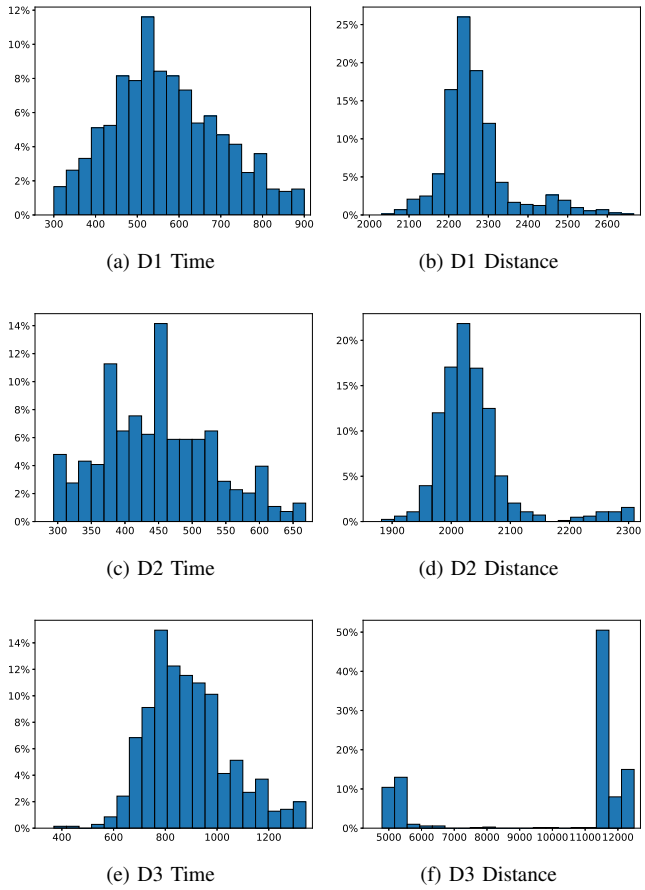


Fig. 5. Interval statistics of trajectory characteristics for three datasets.

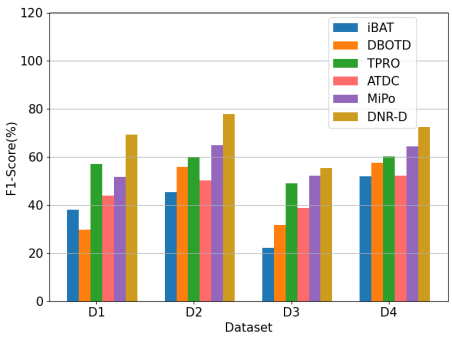


Fig. 6. Comparison of F1 scores for all methods on the original dataset.

all methods that output anomaly scores. Table II represents a performance comparison of all methods on the four original datasets, while Figure 6 shows the performance comparison of F1 scores.

For the quantitative results, DNR-D performs well on all four datasets, with higher precision, recall, and F1 scores than other methods. On the D1 dataset, DNR-D's precision and F1 score are 11.85 and 12.25 percentage points higher than the second-ranked TPRO, respectively, and its recall is 11.27 percentage points higher than the second-ranked ATDC. On the D2 dataset, DNR-D's precision is 11.28 percentage points higher than the second-ranked ATDC, and its recall

TABLE II
PERFORMANCE COMPARISON OF ALL METHODS ON THE ORIGINAL DATASET

Dataset	**D1**			**D2**			**D3**			**D4**		
Metric	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
iBAT	0.368	0.394	0.381	0.402	0.522	0.454	0.191	0.264	0.222	0.398	0.536	0.520
DBOTD	0.289	0.309	0.299	0.494	0.641	0.558	0.274	0.377	0.317	0.512	0.655	0.576
TPRO	0.552	0.591	0.571	0.528	0.686	0.597	0.424	0.584	0.492	0.530	0.654	0.604
ATDC	0.344	0.605	0.438	0.476	0.447	0.504	0.400	0.377	0.388	0.568	0.603	0.522
MiPo	0.500	0.535	0.517	0.574	0.746	0.649	0.452	0.622	0.523	0.582	0.740	0.645
DNR-D	**0.671**	**0.718**	**0.693**	**0.689**	**0.895**	**0.779**	**0.479**	**0.660**	**0.555**	**0.667**	**0.805**	**0.725**

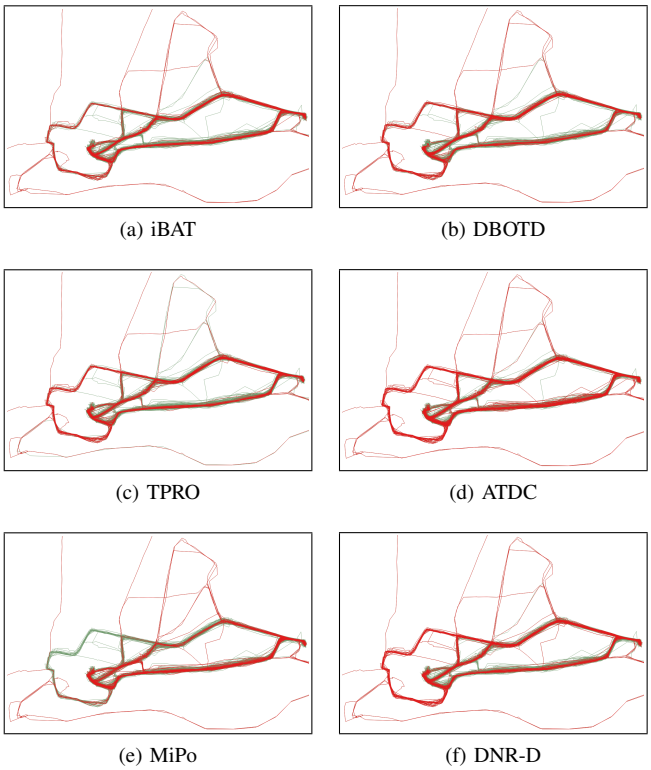
TABLE III
PERFORMANCE COMPARISON OF ALL METHODS ON TEMPORAL ANOMALY DATASET

Dataset	**TD1**			**TD2**			**TD3**			**TD4**		
Metric	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
iBAT	0.107	0.137	0.120	0.124	0.147	0.135	0.079	0.121	0.096	0.104	0.135	0.132
DBOTD	0.181	0.232	0.203	0.156	0.228	0.185	0.108	0.128	0.117	0.167	0.230	0.193
TPRO	0.657	0.793	0.718	0.512	0.756	0.610	0.411	0.583	0.482	0.523	0.768	0.625
ATDC	0.483	0.652	0.555	0.459	0.809	0.586	0.120	0.857	0.210	0.503	0.801	0.564
MiPo	0.194	0.608	0.294	0.211	0.545	0.305	0.185	0.928	0.309	0.223	0.536	0.325
DNR-T	0.485	0.739	0.586	0.487	0.952	0.645	0.538	0.500	0.518	0.530	0.823	0.677

and F1 scores are 14.92 and 12.98 percentage points higher than the second-ranked MiPo, respectively. On the D3 dataset, DNR-D's precision, recall, and F1 score are 2.74, 3.78, and 3.18 percentage points higher than the second-ranked MiPo, respectively. On the D4 dataset, DNR-D's precision, recall, and F1 score are 8.5, 6.5, and 8.0 percentage points higher than the second-ranked MiPo, respectively.

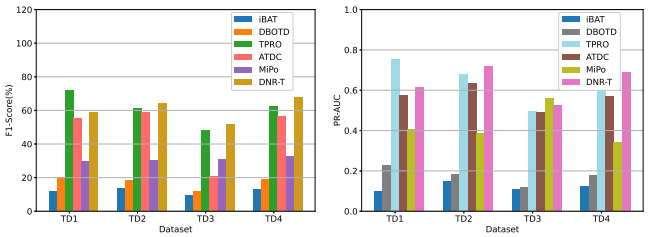
For the analysis of results, DNR-D's strong performance across all datasets is especially notable on the D2 dataset, due to its denser trajectory distribution, which helps organize the vehicular social network and learn better embedding vectors. The performance on the D4 dataset is second only to the D2 dataset for similar reasons. In contrast, the inherent limitations of other methods hinder their ability to effectively handle dense trajectory sets, thereby impacting their performance. Notably, most methods experience a significant decline in performance on the D3 dataset, indicating that the presence of two typical routes with different time periods can significantly impede the judgment ability of these methods. DNR-D and MiPo perform well here due to the LOF detection algorithm, which effectively detects anomalous trajectories within trajectory clusters.

For the visulization of results, Figure 7 illustrates the anomaly detection results on the D1 dataset for each method. The red trajectories indicate the anomalous trajectories detected by the method, while the green trajectories represent the normal ones. The figure presents the evaluation capabilities of each method. Specifically, iBAT, DBOTD, and TPRO identify some anomalous trajectories, but they miss more anomalous trajectories that are related to route switches. On the other hand, ATDC and MiPo can detect more anomalous trajectories related to detours and route switches. Nevertheless, ATDC achieves excellent results at the cost of identifying more normal trajectories as anomalous, which leads to an unbias in recall and precision. As for MiPo, although it obtains excellent results, it completely overlooks the anomalous route switch in the other typical route and focuses on the one with denser trajectories mistakenly. Determining whether route switches with higher trajectory density are anomalous or not based solely on the trajectories is challenging since it represents a route that some vehiculars would consider normal unless the trajectories are generated exclusively by a few vehiculars.



(a) iBAT (b) DBOTD (c) TPRO (d) ATDC (e) MiPo (f) DNR-D

Fig. 7. Visualization of anomaly detection results for all methods on D1 dataset.



(a) F1 Score (b) PR-AUC

Fig. 8. Comparison of F1 scores and PR-AUC for all methods on temporal anomaly dataset.

DNR also identifies some incorrectly anomalous trajectories in typical routes, which explains why its identification results did not achieve higher scores.

Temporal anomaly dataset. For the four temporal anomaly datasets, the proposed method calculates anomaly scores using average transition embeddings, referred to as DNR-T. Trajectories with anomaly scores ranking in the top 5% are considered anomalous trajectories. The performance comparison of all methods on the four temporal anomaly datasets is presented in Table III.

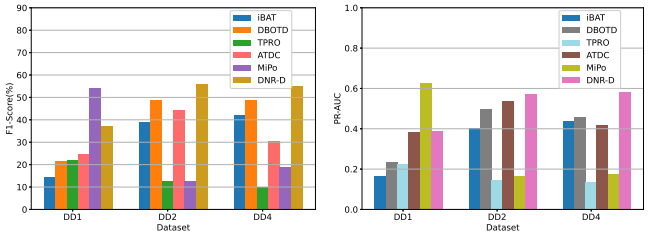
For performance analysis on each dataset, DNR-T ranks among the top methods with high precision, recall, and F1 scores across the four datasets. On TD1, it performs 13.25% worse than the top-performing TPRO. TPRO takes into account the temporal factor of typical routing, while the DNR method focuses on detecting the anomalous trajectories by utilizing the dynamic properties of the vehicular social network, and thus exhibits a weaker performance in the detection of temporal anomalies than TPRO. Nevertheless, DNR-T ranks second in all metrics and demonstrates some ability in detecting temporal anomalies. On TD2, although DNR-T's precision is lower than TPRO, its recall and F1 scores are higher, balancing the trade-off for overall improvement. On TD3 and TD4, DNR-T outperforms TPRO in key metrics. Compared to MiPo and ATDC, our method achieves a relatively balanced result, surpassing other methods in F1 score.

For comprehensive performance analysis across datasets, Figure 8a compares F1 scores across datasets. Overall, DNR-T and TPRO perform similarly well. Most methods show decreased performance on TD3 due to its generation process, which involves manually labeled normal trajectories scaled using quadratic interpolation. The two distinct routes on D3 make manual anomaly detection challenging. Therefore, some trajectories labeled as normal by humans may be considered anomalous by other methods, which can result in the misclassification of the anomalous trajectories on the TD3 dataset. Nevertheless, our method achieves excellent results, validating its ability to detect temporal anomalies.

Figure 8b shows PR-AUC results, highlighting that DNR-T performs at least the second best on each dataset and excels on TD2 and TD4. Although the gaps in performance between the proposed method and the ATDC and MiPo methods are smaller than those observed with the F1 score, the ATDC ranks only third in all four datasets, while the stability of MiPo is inferior.

TABLE IV
PERFORMANCE COMPARISON OF ALL METHODS ON DETOUR ANOMALY DATASET

Dataset	Model	Metric		
		Precision	Recall	F1-Score
DD1	iBAT	0.138	0.147	0.142
	DBOTD	0.211	0.217	0.214
	TPRO	0.218	0.225	0.221
	ATDC	0.155	0.588	0.245
	MiPo	0.397	0.852	0.542
	DNR-T	0.361	0.382	0.371
DD2	iBAT	0.381	0.400	0.390
	DBOTD	0.476	0.500	0.487
	TPRO	0.123	0.125	0.124
	ATDC	0.315	0.750	0.444
	MiPo	0.094	0.200	0.128
	DNR-T	0.547	0.575	0.561
DD4	iBAT	0.360	0.425	0.423
	DBOTD	0.452	0.483	0.489
	TPRO	0.147	0.125	0.990
	ATDC	0.345	0.640	0.307
	MiPo	0.106	0.246	0.188
	DNR-T	0.525	0.508	0.551



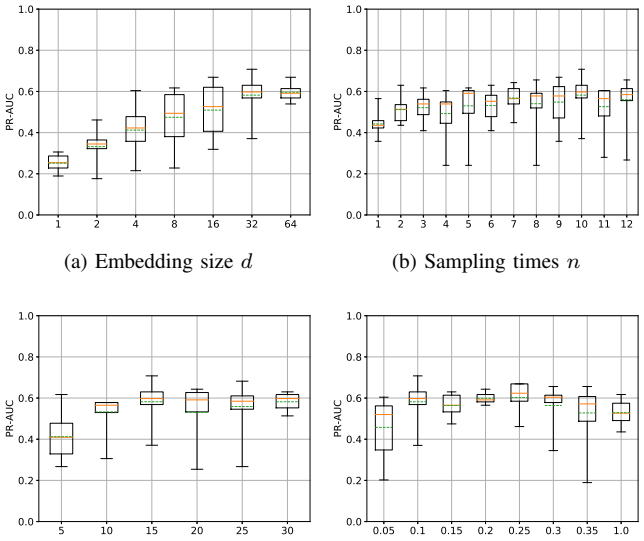
(a) F1 Score (b) PR-AUC

Fig. 9. Comparison of F1 scores and PR-AUC for all methods on detour anomaly dataset.

Detour anomaly dataset. In these datasets, our method calculates the anomaly score using driving embedding, i.e., DNR-D. Trajectories with anomaly scores ranking in the top 5% are considered anomalous. Table IV shows the performance comparison of all methods on both detour anomaly datasets.

For performance analysis on each dataset, due to issues with the D3 dataset mentioned earlier, we only represent experimental results for the three detour anomaly datasets. Comparing the results on these datasets, DNR-D performs either first or second in most metrics, with a high overall ranking. On DD1, DNR-T's precision is lower than MiPo, while its recall is ranked third, resulting in an F1 score that ranks second overall. On DD2, DNR-T has 7.14% higher precision than DBOTD and achieves the highest F1 score. On DD4, it outperforms DBOTD in precision, recall, and F1 scores by 7.30%, 2.52%, and 6.21%, respectively.

For comprehensive performance analysis across datasets, we compared the performance of different methods on the detour anomaly dataset using the F1 score and PR-AUC metric, as shown in Figure 9a and Figure 9b. DNR-D ranks among the top performers on three datasets for the F1 score, achieving the best performance on the DD2 dataset. In contrast, MiPo achieves the best performance on the DD1 dataset but performs almost the worst on the DD2 dataset and DD4 dataset. For PR-



(a) Embedding size d (b) Sampling times n

(c) Walk length l (d) Selection ratio α

Fig. 10. Parameter performance comparison of DNR.

AUC, DNR-D's performance is at least second-best on each dataset, achieving the best performance on the DD2 dataset. The ATDC's performance is always very close to DNR-D, while the performance of MiPo varies greatly between the two datasets, similar to what is observed in the temporal anomaly dataset, indicating its instability.

E. Parameter Study

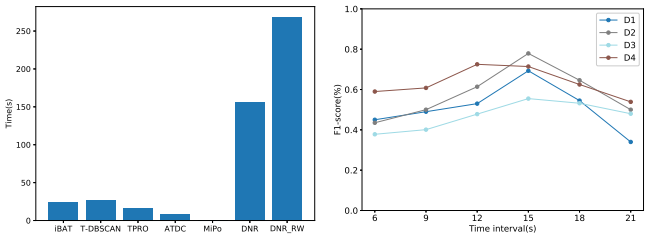
The parameters used in this method consist of the network partitioning parameter α, sampling parameters n and l, and the embedding size d of the neural network. To explore the impact of these parameters, we conduct 10 experiments with different settings for each parameter on the D1 dataset and visualize the results using box plots, as shown in Figure 10. Note that the x-axis of the figure is not uniform for Figure 10a and 10d. In the box plot section, the upper and lower caps represent the maximum and minimum values, respectively. The box represents the middle 50% of the data after excluding the top 25% and bottom 25%. The orange solid line represents the median, and the green dashed line represents the mean.

Embedding size d. Small embedding sizes result in poor performance. As d increases, performance improves, peaking at d=32 with stable results. At d=64, stability increases, but excellent results decrease. Thus, The embedding size of d=32 is chosen for its balance between capturing detailed trajectory differences and computational efficiency.

Sampling times n. The box plot's lower position indicates stable but poor results at low sampling times. Performance improves with increased sampling time, becoming optimal and stable at a sampling time of 10. However, there is some randomness in multiple experiments. The method's instability with certain parameter settings is due to the random walk algorithm, which can cause context information loss for some nodes. Our menthod sets the sampling time for each node is set to 10.

Walk length l. The method's performance generally improves as the walk length increases. Based on the median and box plot, the method tends to stabilize when the walk length reaches 15. Although the mean exhibit significant fluctuations when the walk length is 20, the box plot remains at a high position. Taking into account the occasional variability of neural network methods, it can be assumed that the performance of the method does not deviate significantly. Therefore, setting the walk length to a value greater than or equal to 15 is reasonable, and our method sets this parameter to 15.

Selection ratio α. When the selection ratio is small, the method performs poorly. As it reaches 0.1, performance stabilizes, and further increases in the ratio show little change. At a ratio of 1.0, performance decreases as the dynamic walk algorithm degenerates into a standard random walk algorithm. Therefore, sampling a portion of the nodes is more effective in improving the method's performance for anomaly trajectory detection in dynamic social networks. For parameter selection, if only the current experimental results are considered, setting the parameter to 0.1, 0.2, or 0.25 is beneficial. Specifically, the highest-performing method is obtained when the parameter is set to 0.1, the most stable method is obtained when the parameter is set to 0.2, and the overall best performance is obtained when the parameter is set to 0.25. However, our method sets the selection ratio to 0.1 in order to pursue the highest results.



(a) Running time (b) time interval

Fig. 11. Running time of each method on D1 dataset and time interval comparison of DNR.

Time interval. Each trajectory includes GPS coordinates reported every 15 seconds by the taxi in the Porto dataset. When converting trajectory points to grid numbers, we notice that even though it is a 15 seconds time granularity, the grid numbers where neighboring trajectory points are located will still be the same, so the 15 seconds time granularity is not considered coarse. As shown in the figure 11b, 15 seconds is a more appropriate time granularity in the Porto dataset. And in the Chengdu dataset, 12 seconds is the most appropriate time interval, and any coarser time granularity will lead to the inability to detect some anomalies, which makes the performance of anomaly detection degraded. In the study of vehicular social networks, a proper temporal segmentation can provide enough precision to capture dynamic changes that occur in the short term without being too sensitive. This fine-grained temporal segmentation can help detect anomalous behaviors occurring in the short term while reducing false alarms for normal behaviors.

Running time and Complexity Analysis of DNR. We also explore the running time of all methods on the D1 dataset. DNR represents our method using the dynamic walk algorithm, while DNR_RW represents a random walk algorithm that samples all nodes. The results, shown in Figure 11a, demonstrate that the use of neural networks in DNR leads to a significantly longer learning time compared to the statistical-based methods. However, DNR utilizing sampling nodes before context sampling reduces the time required for both sampling and learning representations, compared to DNR_RW. This suggests that the proposed dynamic walk algorithm is indeed superior to the random walk algorithm, but its major drawback is still the running time. Therefore, running time will be one of the future research directions. In the following we analyze the time complexity of the algorithm. The construction time of the dynamic vehicular social network and representation learning is negligible and the dynamic walk is divided into network partitioning, dynamic network node selection and sampling methods. The time complexity of network partitioning is $O(V+E+V\cdot\log V+k\cdot V)$, the overall time complexity of the dynamic graph node selection algorithm is $O(V)$, where k is the number of iterations, E is the number of edges, and V is the number of nodes. The overall time complexity of the sampling method with wander length L and number of subgraphs P is $O(P\cdot L)$. Hence, the overall complexity of trajectory encoding is $O(E+V\cdot(\log V+k)+P\cdot L)$.

F. Case Study

In this section, the focus will be shifted from quantitative results back to the method itself. First, the rationale for constructing a dynamic vehicular network will be explored. Figure 12 shows the grid heatmap of vehicular movements over time in the D1 dataset. Grids with more vehiculars have higher brightness. By generating heatmaps at different times, the approximate trajectories of vehiculars can be obtained.

In the figure, vehiculars depart from the lower right corner of the map at $t = 3$ and arrive at their destination at $t = 43$. Two different typical routes are taken by vehiculars during their journey: the upper route has a higher density of vehiculars with more bright grids, while the lower route is still considered normal but has fewer bright grids. At $t = 13$, there are four additional typical trajectories besides the typical routes: fast, medium, and slow trajectories on the upper route, and one trajectory on the lower route, as indicated by the four high-brightness grids. Typical trajectories have more temporal features than typical routes, enabling them to better reflect the spatiotemporal properties of trajectories and identify anomalous trajectories from both temporal and spatial aspects. Specifically, at $t = 8, 18, 23, 43$, some grids have low vehicular density, suggesting the possibility of abnormal driving behavior such as detours, speeding, and slow driving, resulting in spatiotemporal abnormalities.

To demonstrate our claim, we filter all anomalous trajectories detected by DNR in the D1 dataset to select those that could be identified correctly by DNR but not by others. Figure 13 displays one such trajectory in red. At a glance, the trajectory appears to follow a typical route without any

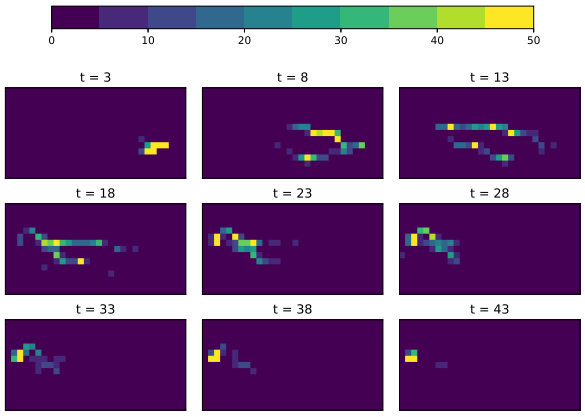


Fig. 12. Visualization of anomaly detection results for all methods on D1 dataset.

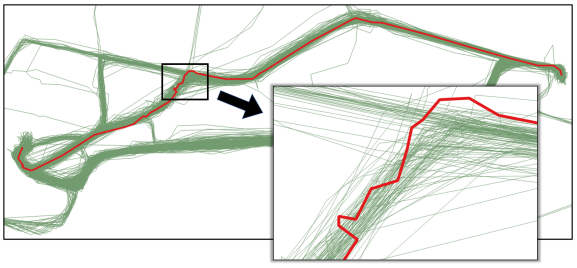


Fig. 13. Visualization of anomaly detection results for all methods on D1 dataset.

spatial anomalies such as detours or path switching. However, upon closer inspection, it exhibits highly unusual durations at intersections. Specifically, vehiculars in this dataset report coordinate every 15 seconds, yet this trajectory has three straight segments at the intersection and six straight segments after the intersection, indicating that it takes 45 seconds to pass the intersection and then 1 minute and 30 seconds to continue driving without significant stopping. In contrast, other normal trajectories can pass through this section quickly. Therefore, although this trajectory belongs to a typical route, it is clearly an anomalous trajectory deviating from typical trajectories compared to other normal trajectories.

V. Future Directions

However, the proposed method has certain limitations. While it effectively identifies spatio-temporal anomalies, but it maybe fall short in capturing hidden dynamic network changes and runtime. In the future, our research direction is to optimize the dynamic walk and running time.

To capture hidden dynamic changes in the network, we will focus on multi-scale analysis and contextual information integration. Analyzing node selection and topology changes at different time scales allows detection of both short-term bursts and long-term trends. Integrating external factors like traffic rule changes and weather conditions enhances the understanding of implicit dynamic network changes. This contextual information helps to explain unexpected events and periodic changes in the network and provides richer contextual

information, thus improving the accuracy and robustness of anomaly detection. The combination of multi-scale analysis and contextual information integration enables our approach to capture and understand the complex evolutionary features of dynamic networks more comprehensively. To improve anomaly detection, we can balance local and global information sampling by using adaptive strategies. These strategies adjust node sampling based on network changes. The sampling of local and global information can be balanced when the evolution of the global network structure is more significant. And diverse node sampling strategies are designed to ensure that the algorithm can cope with different situations, which can reduce the instability of anomaly detection results due to sampling bias.

The running time can be reduced by improving the efficiency of node selection, e.g., improving node selection based on topological changes in the network structure, i.e., improving the scoring function for assessing the importance of nodes. Improving the sampling efficiency, e.g., learning the jump time step, which improves the time but does not radically reduce the computational complexity and may also reduce the stability of the method. It is also possible to parallelize the two key steps of dynamic network node selection and sampling to improve the operational efficiency of the algorithm.

VI. Conclusion

Existing anomaly trajectory detection methods primarily focus on spatial anomalies, overlooking temporal anomalies. In this paper, we introduce DNR, a novel unsupervised method that integrates the latest advancements in network research into trajectory analysis. DNR constructs a dynamic vehicular social network and employs incremental learning to embed it into two latent representations. Experimental results on manually labeled and anomaly-injected datasets establish the superiority and stability of DNR compared to baseline methods. Moreover, the findings reveal DNR's capability to detect deviations that elude other methods.

Acknowledgments

The authors would like to thank Hang Lin and Juntao Wang for their assistance in dataset preprocessing and anomaly injection.

References

[1] X. Kong, J. Wang, Z. Hu, Y. He, X. Zhao, and G. Shen, "Mobile trajectory anomaly detection: Taxonomy, methodology, challenges, and directions," *IEEE Internet of Things Journal*, 2024.

[2] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "Ibat: Detecting anomalous taxi trajectories from gps traces," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, New York, NY, USA, 2011, p. 99–108.

[3] Z. Lv, J. Xu, P. Zhao, G. Liu, L. Zhao, and X. Zhou, "Outlier trajectory detection: A trajectory analytics based approach," in *Database Systems for Advanced Applications*, Cham, 2017, pp. 231–246.

[4] J. Wang, Y. Yuan, T. Ni, Y. Ma, M. Liu, G. Xu, and W. Shen, "Anomalous trajectory detection and classification based on difference and intersection set distance," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2487–2500, 2020.

[5] X. Kong, B. Zhu, G. Shen, T. C. Workneh, Z. Ji, Y. Chen, and Z. Liu, "Spatial-temporal-cost combination based taxi driving fraud detection for collaborative internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3426–3436, 2022.

[6] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, and Z. Wang, "iboat: Isolation-based online anomalous trajectory detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 806–818, 2013.

[7] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Time-dependent popular routes based trajectory outlier detection," in *Web Information Systems Engineering – WISE 2015*, Cham, 2015, pp. 16–30.

[8] J. Mao, T. Wang, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2696–2709, 2017.

[9] J. Yang, X. Tan, and S. Rahardja, "Mipo: How to detect trajectory outliers with tabular outlier detectors," *Remote Sensing*, vol. 14, no. 21, 2022.

[10] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: Long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[11] Y. Cheng, B. Wu, L. Song, and C. Shi, "Spatial-temporal recurrent neural network for anomalous trajectories detection," in *Advanced Data Mining and Applications*, Cham, 2019, pp. 565–578.

[12] Y. Liu, K. Zhao, G. Cong, and Z. Bao, "Online anomalous trajectory detection with deep generative sequence modeling," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, Dallas, TX, USA, 2020, pp. 949–960.

[13] K. Kumaran Santhosh, D. P. Dogra, P. P. Roy, and A. Mitra, "Vehicular trajectory classification and traffic anomaly detection in videos using a hybrid cnn-vae architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 891–11 902, 2022.

[14] C. Wang, K. Li, and L. Chen, "Deep unified attention-based sequence modeling for online anomalous trajectory detection," *Future Generation Computer Systems*, vol. 144, pp. 1–11, 2023.

[15] Q. Zhang, Z. Wang, C. Long, C. Huang, S.-M. Yiu, Y. Liu, G. Cong, and J. Shi, "Online anomalous subtrajectory detection on road networks with deep reinforcement learning," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 246–258.

[16] X. Kong, W. Zhou, G. Shen, W. Zhang, N. Liu, and Y. Yang, "Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data," *Knowledge-Based Systems*, vol. 261, p. 110188, 2023.

[17] X. Kong, H. Lin, R. Jiang, and G. Shen, "Anomalous sub-trajectory detection with graph contrastive self-supervised learning," *IEEE Transactions on Vehicular Technology*, 2024.

[18] L. Xie, T. Guo, J. Chang, C. Wan, X. Hu, Y. Yang, and C. Ou, "A novel model for ship trajectory anomaly detection based on gaussian mixture variational autoencoder," *IEEE Transactions on Vehicular Technology*, 2023.

[19] C. D. T. Barros, M. R. F. Mendonça, A. B. Vieira, and A. Ziviani, "A survey on embedding dynamic graphs," *ACM Comput. Surv.*, vol. 55, no. 1, nov 2021.

[20] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2015, p. 1365–1374.

[21] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Effective and efficient trajectory outlier detection based on time-dependent popular route," *World Wide Web*, vol. 20, no. 1, pp. 111–134, 2017.

[24] R. Jiao, J. Bai, X. Liu, T. Sato, X. Yuan, Q. A. Chen, and Q. Zhu, "Learning representation for anomaly detection of vehicle trajectories," *arXiv preprint arXiv:2303.05000*, 2023.

[25] A. Radulescu, Y. S. Shin, and Y. Niv, "Human representation learning," *Annual Review of Neuroscience*, vol. 44, no. 1, pp. 253–273, 2021.

[26] H. Iuchi, T. Matsutani, K. Yamada, N. Iwano, S. Sumi, S. Hosoda, S. Zhao, T. Fukunaga, and M. Hamada, "Representation learning applications in biological sequence analysis," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 3198–3208, 2021.

[27] H. F. Yang, J. Cai, C. Liu, R. Ke, and Y. Wang, "Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning," *Transportation Research Part C: Emerging Technologies*, vol. 148, p. 103982, 2023.

[28] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.

[29] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2014, p. 701–710.
[30] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016, p. 855–864.
[31] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, Republic and Canton of Geneva, CHE, 2015, p. 1067–1077.
[32] C. Hou, H. Zhang, S. He, and K. Tang, "Glodyne: Global topology preserving dynamic network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4826–4837, 2022.
[33] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, *Recent advances in graph partitioning*. Springer, 2016.
[34] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.
[35] M. Hou, J. Ren, D. Zhang, X. Kong, D. Zhang, and F. Xia, "Network embedding: Taxonomies, frameworks and applications," *Computer Science Review*, vol. 38, p. 100296, 2020.
[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, vol. 26, Lake Tahoe, Nevada, United States, 2013, pp. 3111–3119.
[37] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2000, p. 93–104.
[38] G. Zheng, S. L. Brantley, T. Lauvaux, and Z. Li, "Contextual spatial outlier detection with metric learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2017, p. 2161–2170.

Guojiang Shen received the B.S. degree in Control Theory and Control Engineering and the Ph.D. degree in Control Science and Engineering from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively. He is currently a Professor in the College of Computer Science and Technology, Zhejiang University of Technology. His current research interests include artificial intelligence, Big Data analytics and intelligent transportation systems.

Yuwei He received the B.Sc. degree from China Jiliang University,Hangzhou, China, in 2022, and currently studying for a master's degree at Zhejiang University of Technology, Hangzhou, China. Her main research interests include urban computing and data mining.

Xiangjie Kong (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Full Professor with College of Computer Science and Technology, Zhejiang University of Technology. Previously, he was an Associate Professor with the School of Software, Dalian University of Technology, China. He has published over 200 scientific papers in international journals and conferences (with over 170 indexed by ISI SCIE). His research interests include network science, mobile computing, and urban computing. He is a Senior Member of the IEEE and CCF and is a member of ACM.

Zhanhao Ji received the B.S. degree in software engineering from Zhejiang University of technology, China, in 2020. He is currently working towards the M.S. degree in the College of Software, College of Computer Science and Technology, Zhejiang University of technology, China. His research interests include urban data mining and intelligent transportation system.

Lei Wang (Member, IEEE) received the B.E. degree in automation, the B.S. degree in applied mathemat ics, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2004, 2004, and 2009, respectively.

From 2014 to 2015, he was a Senior Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, and as a Senior Research Associate Professor with the Department of Mechanical Engineering, from 2015 to 2016. He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His current research interests include modeling, control, and optimization in complex networked systems.

Dr. Wang is currently the Associate Editor of the ISA Transactions (USA).

Junjie Zhou received the master's degree in Control Engineering from Zhejiang University, Hangzhou, China, in 2016. He is currently a Senior Engineer and Vice President of Zhejiang Supcon Information Co., Ltd., Hangzhou. His research interests include traffic signal control, big data analytics, and intelligent transportation system.

Linglong Hu received the master's degree in Control Science and Engineering from Zhejiang University, Hangzhou, China, in 2014. She is currently a senior engineer of Zhejiang Supcon Information Co., Ltd., Hangzhou. Her research interests include artificial intelligence, cube dynasim, and intelligent transportation system.